Image Webs: Computing and Exploiting Connectivity in Image Collections

Kyle Heath[1], Natasha Gelfand[2], Maks Ovsjanikov[1], Mridul Aanjaneya[1], Leonidas J. Guibas[1]
Stanford University[1], Nokia Research - Palo Alto, CA[2]

{heathkh,maks,aanjneya}@stanford.edu, guibas@cs.stanford.edu, natasha.gelfand@nokia.com

Abstract

The widespread availability of digital cameras and ubiquitous Internet access have facilitated the creation of massive image collections. These collections can be highly interconnected through implicit links between image pairs viewing the same or similar objects. We propose building graphs called Image Webs to represent such connections. While earlier efforts studied local neighborhoods of such graphs, we are interested in understanding global structure and exploiting connectivity at larger scales. We show how to efficiently construct Image Webs that capture the connectivity in an image collection using spectral graph theory. Our technique can link together tens of thousands of images in a few minutes using a computer cluster. We also demonstrate applications for exploring collections based on global topological analysis.

1. Introduction

The widespread availability of digital cameras and ubiquitous Internet access have helped create massive personal and public image collections on sites such as *Facebook* and *Flickr*. Images in these large collections can be highly correlated because there are many images of the same or similar objects under different viewing conditions. When an image collection is sufficiently well interlinked by shared objects, it becomes interesting to study the structure of the network formed by such links and to exploit the connections among images.

We propose building graphs called *Image Webs* to represent the connections between images in a collection. Each node in an Image Web corresponds to a region of connected pixels in an image. Edges in the Image Web connect regions in different images that are similar under an affine transform. These regions are extracted using a process called *Affine Cosegmentation* that we describe in Section 3 to produce match links. There are also edges connecting distinct regions that occur in the same image.

Evidence from cognitive neuroscience suggests that the brain may learn representations of objects as a continuous
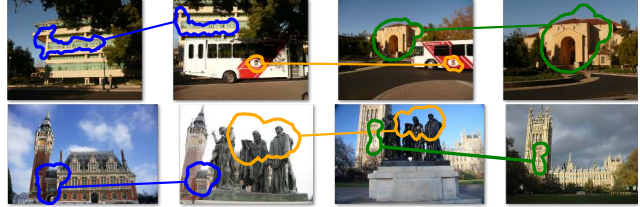

Figure 1. Image Webs can connect images of buildings along a bus route via the moving bus (top row) or images of Calais and London through casts of the Rodin sculpture "The Burghers of Calais", which commemorates a historical connection between the two cities in the Hundred Years' War (bottom row).

manifold in a high-dimensional space from a large number of observations of objects under viewpoint and illumination changes [7]. We believe Image Webs may be a promising tool for many vision problems because they provide a similar representation – a discrete sampling of an image manifold produced by linking views of an object with sufficiently similar pose and illumination conditions.

As we briefly survey in Section 2, considerable research has already been devoted to exploiting match links to detect images of popular landmarks [25] and to annotate them [11], or for 3-D reconstruction of landmarks [23]. These applications focus on isolated and localized clusters within an Image Web. Instead, we are interested in relationships between images, which may be visually quite different, arising from many transitive connections. For example, in the top row of Figure 1, images of two different buildings at a university are connected by a campus bus that frequently stops near each building. In the bottom row, images of the town hall of Calais (France) and the Palace of Westminster (London) are connected because a cast of Rodin's sculpture "The Burghers of Calais" appears in front of both buildings. While prior work has mostly used match links to connect images of a particular static scene, this small example shows that transitive "non-local" associations can also be quite interesting. Non-local connections can be established through mobile objects that co-visit static objects, through objects that have multiple instances in the world with similar visual appearance, or even through static objects connecting mobile objects (e.g. people visiting the

same monuments).

The focus of this paper is *to study the large-scale connectivity of image collections and the use of this connectivity in a variety of applications*. Our contributions are twofold. First, a technique for building Image Webs efficiently is presented in Section 3. Reliably detecting matching regions between images is an expensive step using current feature matching techniques. To handle collections of tens of thousands to millions of images, we simply cannot afford the quadratic cost of directly comparing every pair of images. Since our applications are concerned with global connectivity of the Image Web, we want the construction process to correctly capture the entire structure, and not spend a disproportionate amount of time inter-connecting already densely-linked regions. Towards this end, we propose a novel approach based on spectral graph theory and Content Based Image Retrieval (CBIR) that chooses to test image pairs that are likely to globally improve a spectral measure of graph connectivity. Given a limited budget of computation time or number of edges that can be stored, our algorithm will capture significantly more of the global structure of the Image Web than an approach based on CBIR scores alone.

Our second contribution consists of several new ways of exploring image collections by exploiting the connectivity captured in an Image Web (Section 4). We first present a visual-hyperlink browser, which allows users to navigate in the image collection using linked regions between images like hyperlinks in web pages. Because an Image Web can be large and complex, a global summary can be useful to visualize its essential structure. Towards this end we use tools from persistent homology and cohomology in computational topology to extract junctions (branch points) and form a global map of the Image Web, as well as to parametrize images along linear or circular structures — which together greatly improve the navigation experience and understanding of the structure of the Image Web. We also show how computing paths between densely connected regions of the web can reveal interesting relationships between images. These paths are usually ignored in previous clustering-based applications, but are informative because they summarize how densely connected regions of the web are inter-related.

2. Related work

The first step of many methods for organizing, annotating, and browsing large image collections is to discover pairs of matching images using local feature matching techniques such as [17], and to organize these matches into a graph structure. In this section, we will review some of the previous methods for constructing and using different image graphs.

An intuitive method for browsing many images of the same scene was proposed in [23]. This approach uses pairwise image matching to construct a graph of images, followed by a structure from motion (SfM) computation to recover the camera poses and a sparse 3D reconstruction of the scene. The user can then browse the collection in interesting ways using the recovered spatial relationships between images. In [22], subsets of images corresponding to panoramas and fly-arounds are detected by analyzing the camera poses and fields of view of all the images of a scene.

Graphs of matching images or image regions are used in various image clustering and labeling applications. In [21], clustering is performed in the image graph to choose a set of representative images for an image collection. The approach of [4] establishes initial matches between images using min-hashing, and then uses query expansion to find clusters of similar images for landmark discovery and 3D reconstruction. In [11], labels are transferred across match links in an image graph for landmark identification in holiday photos. A graph structure similar to the Image Webs we use was proposed in [25]: the graph is built on image regions consisting of groups of matching features, with edges linking regions that match in different images, as well as regions that co-appear in a single image. This structure is then used as an input to a clustering algorithm that clusters and labels photos of landmarks in a large collection.

Link analysis techniques developed for indexing web pages have been applied to images. In [14], the PageRank algorithm is run on a graph built from images returned by Google Product Search to re-rank and group similar products together. In [15], a lower-level structure is built with graph nodes corresponding to individual local features, with edges linking features sharing a similar descriptor and passing a geometric consistency test. Spectral clustering is then performed on the graph for unsupervised learning of object categories.

The Image Web we use is similar to some of the graphs described above [25, 4], but is built to capture different structure in the collection. The clustering, labeling, and object detection applications mentioned above generally aim to partition an image graph into components that correspond to different landmarks, objects, or labels. In this work, we are instead interested in the global structure of the connectivity of a set of images and long paths in the Image Web. A related problem of connecting pairs of images through a collection was examined in [22], however, in that work the discovered paths are restricted to paths in the physical space of camera poses of the same scene. In our work, we use a more general notion of a path, which connects images through shared regions and objects and apply tools from computational topology to summarize the structure. As a result, we are still able to connect images when the number of views is insufficient for 3-D reconstruction or, more interestingly, when the objects are not part of the same scene. Such non-

physical paths can have informative semantic meaning, as demonstrated by the examples in Figure 1.

For large datasets, the quadratic cost of pairwise image matching is impractical. Content-Based Image Retrieval (CBIR) techniques based on Bag-of-Visual-Words models [13], global image descriptors [16] or geometric min-hashing [5] have been used to quickly filter out many images that are unlikely to match. These accelerations may exclude some potentially relevant images. To mitigate this problem, query expansion [6] has been used recover missing matches by re-querying with each of the images returned from an initial query. In [1], Agarwal et al. use a combination of CBIR, query expansion, and parallelization to do 3D reconstruction from image collections with hundreds of thousands of images. Philbin et al. [20] use CBIR and geometric consistency checks to build graphs on very large image collections of urban environments. Their target application is clustering, so the construction focuses on densely-connected regions of the graph and the clustering specifically aims at cutting links that bridge gaps across clusters. In contrast, our construction and navigation algorithms aim at finding and exploring such links.

We believe that connectivity is an important property for image graphs that has not been explored in existing work and would lead to improvements in many of the aforementioned applications. For example in [1], the authors show that their 3D reconstruction technique is forced to split a large 3D model into two smaller ones at a point of low connectivity in the initial image graph. Our graph construction method specifically aims at improving connectivity around such bottlenecks.

3. Image Web Construction

The basic entity in an Image Web is a *region*, which corresponds to a connected component of pixels in the image. We discover corresponding regions between images using a process called affine cosegmentation (or just cosegmentation for short), described in the next section.

Match links are added between the matching regions in different images generated by cosegmentation. Additional links are introduced between regions in the same image to represent their co-occurrence in the same scene, or their shared identity if they overlap.

3.1. Affine Cosegmentation

The affine cosegmentation process takes a pair of images (I_a, I_b) as input and detects regions of maximal size in image I_a that correspond under an affine transformation to regions in I_b. The approach, illustrated in Figure 2, uses local-feature matching techniques and works as follows.

First, affine covariant local features are extracted from the images. Harris-affine and Hessian-affine keypoint
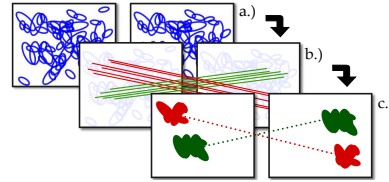

Figure 2. Affine cosegmentation is performed by a.) extracting local features b.) detecting affine consistent feature matches c.) extracting regions by merging keypoint support regions.

detectors [18] and the Maximally Stable Color Regions (MSCR) keypoint detector [10] are used in order to detect keypoints on both smooth and textured image regions. Keypoints are described using the SIFT descriptor. Next, tentative correspondences between local features are generated by matching each feature in one image to its approximate nearest neighbor in the other (in descriptor space). Matching is performed for each of the three keypoint types using Lowe's ratio test criterion [17] to select candidate matches. In practice, most of these tentative correspondences are incorrect because of variation in the set of selected keypoints, large changes in viewpoint and lighting, the use of approximate nearest neighbor algorithms to accelerate matching, and the existence of multiple instances of the same feature in an image.

To deal with incorrect matches, the RANdom Sample Consensus algorithm (RANSAC) [9] is iteratively invoked to detect sets of geometrically consistent feature matches. We apply RANSAC to find a maximal set of feature matches such that features in one image can be mapped to their corresponding features in the other by an affine transformation. The features belonging to a maximal set are removed from consideration and the process is iterated until no sufficiently large set of features can be found. The geometric constraint is effective at filtering out the large number of incorrect matches. Finally, the point correspondences are converted to a region-based representation of the extent of the shared region in each image. The union of the ellipse-shaped affine-covariant keypoint regions is computed and the boundary of its largest connected region is used as a rough segmentation of the shared region. For all our examples, we compute cosegmentation on images scaled to 0.3 megapixels and consider a region match successful if it contains 10 feature matches.

3.2. Efficient construction

Ideally, an Image Web should be built by attempting cosegmentation on all pairs of images in the collection. In practice, all-pairs matching is too expensive and also highly inefficient since only a small fraction of pairs actually match in a large collection. To address this problem, we have developed a process for selecting pairs of images for cosegmentation so that we can quickly recover the essential structure that would result from all-pairs matching with a rela-

tively small number of cosegmentation operations.

We define an image-graph to be a graph on images where an edge (I_1, I_2) between images indicates that cosegmentation has succeeded in matching regions between the two images, equivalent to the match-graph used in [1]. A pair of images (I_1, I_2) for which cosegmentation has not yet been performed represents a potential edge in the image-graph which we call a cosegmentation candidate. The Image Web construction proceeds in two phases. The goal in the initial phase is to quickly discover the connected components of the image-graph that would exist if all-pairs matching were performed. The goal in the second phase is to add edges within each connected component to capture the same global connectivity that would result from all-pairs matching.

Phase 1: Discovering connected components

Cosegmentation candidate selection in Phase 1 is guided by Content Based Image Retrieval (CBIR) using the Bag-of-Visual-Words model. Given a query image, CBIR quickly generates a ranked list of other images that contain similar local features. We use the CBIR technique of [13].

Our proposed candidate selection strategy in Phase 1 is as follows. For each image, a CBIR query returns the top k most similar images and a corresponding similarity score. Each image is paired with its k most similar images and all such pairs are sorted by their associated CBIR similarity score. Candidate pairs are considered in order of decreasing similarity score so that the candidate pairs most likely to match are given priority. We impose a further restriction that candidates with images in the same connected component of the evolving image-graph are skipped. This condition focuses the exploration process on those edges that could potentially merge two connected components and results in connected components that are trees. Phase 1 matching proceeds until the frequency of component merges falls below a threshold or until a fixed budget of cosegmentation operations is exceeded.

Phase 2: Increasing connectivity

Since the strategy in Phase 1 results in a very sparse representation of the connected components, the goal of Phase 2 is to discover additional edges that would increase the connectivity within the components. A robust notion of connectivity used in spectral graph theory is called *algebraic connectivity* and is measured by the second smallest eigenvalue of the graph Laplacian. This eigenvalue is known to relate to the ability of the graph to diffuse information, the convergence rate of random walks, and many other measures of graph connectivity. In our setting, given a connected component discovered in Phase 1, we represent it as

a connected graph $G = (V, E)$ and construct the Laplacian matrix L as follows:

$$
L_{i,j} = \begin{cases} d(i) & \text{if } i = j \\ -1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} ,
$$

where $d(i)$ is the degree of vertex i. The *algebraic connectivity* of G is λ_2, the second smallest eigenvalue of L. Note that L is symmetric and thus has real eigenvalues. In addition the eigenvalues are always positive, and the smallest eigenvalue will always be 0 with a corresponding constant eigenvector. The eigenvector of L that corresponds to λ_2 is called the *Fiedler vector*, and we denote it by v_2.

Ideally, given a graph G we would like to find a fixed number of additional edges that would lead to the highest increase in algebraic connectivity. However, this problem is NP-hard [19]. We adopt an inexpensive greedy strategy proposed by Wang and Piet [24] which produces excellent results in practice. We call this procedure *EdgeRank*, since it allows us to rank potential edges according to their connectivity importance using a power-iteration method similar to the *PageRank* algorithm [3].

We create a set of edge candidates by pairing each image in the component with its top M CBIR matches (we use $M = 25$). For every new edge candidate $e = (s, t)$, we compute its EdgeRank score $c_e = |v_2(s) - v_2(t)|$, i.e. the absolute difference of the values of s and t in the Fiedler vector v_2. We consider adding potential edges by attempting cosegmentation of corresponding images in decreasing order of c_e. Intuitively, this procedure will first suggest edges between pairs of vertices of G, which are not tightly connected. This means that connections across different cliques in the graph will be discovered much quicker than if the CBIR score was used to rank the edges.

Once a cosegmentation attempt is successful and an edge is added to the graph, we need to update the Laplacian L and its Fiedler vector v_2. The latter step can be done by using the previous estimate of v_2 as an initial guess for a power iteration in the subspace orthogonal to the space spanned by the constant eigenvector. In other words, we update v_2 by iterating over:

$$
v_2 = \frac{(2nI - L)v_2}{\|(2nI - L)v_2\|_2}, \quad v_2 = v_2 - \frac{1}{n}\sum_{j=1}^{n} v_2(j),
$$

where n is the number of vertices in G and I is the identity matrix. Note that the first step is simply a sparse matrix-vector multiplication, which can be easily parallelized. In the second step we subtract the mean from all elements of v_2. This ensures that v_2 remains orthogonal to the constant eigenvector, and thus the power iteration will converge to the Fiedler vector. Note that although in general the power iteration is known to be slow, convergence is fast in our set-
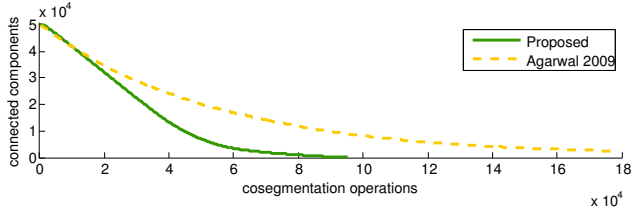
Figure 3. Comparison of Phase 1 strategies on the Pittsburgh collection.

ting since we do not expect the Fiedler vector to change drastically after one edge addition.

Using this method we obtain a more well-connected graph using fewer cosegmentation attempts and generate a smaller number of edges, thus increasing sparsity. We evaluate these improvements on real data in Section 3.3.

3.3. Evaluation

To evaluate the proposed image-graph construction method, we compare our strategy with the strategy in [1]. We briefly review their strategy here. For discovering connected components (Phase 1), their method first matches each image with all its top k_1 CBIR candidates and in a second stage, it matches each image with its next k_2 best CBIR candidates so long as they lie in different components. To increase the density of matches within a component (Phase 2), their method performs query expansion by selecting each image's 2-hop neighbors as match candidates. In experiments, they set k_1 and k_2 to 10 images and performed 4 rounds of query expansion. We use these same parameters in the following evaluation.

To evaluate Phase 1 construction strategies, we used a collection of 50,224 Google street-view images from downtown Pittsburgh, PA. We constructed an Image Web using our proposed Phase 1 candidate selection strategy (with $k = 25$) and the strategy from [1]. The results are shown in Figure 3. Our proposed strategy converges more quickly to a small number of components, resulting in 107 (23 non-singleton) components after 95,075 cosegmentation operations compared with 2488 (372 non-singleton) components after 177,671 cosegmentation operations.

To evaluate Phase 2 construction strategies, we used a smaller collection of 1,257 images of an art museum. We choose a smaller set so that we can easily visualize the resulting image graphs as shown in Figure 5. The proposed Phase 1 construction method was used to generate an initial graph to be refined. Phase 2 construction was performed with the query expansion strategy from [1] with 4 rounds of expansion, the proposed EdgeRank strategy with $k = 25$, and a related strategy we label CBIR. The CBIR strategy chooses among the same candidates as EdgeRank but ranks candidates by CBIR score instead of connectivity score.
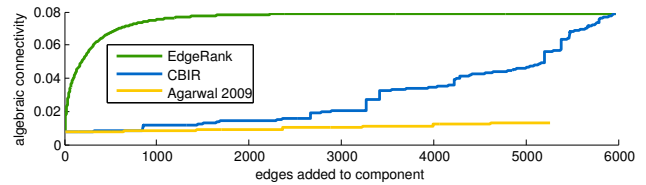
Figure 4(a) shows the relationship between the connectivity and computation time for the three strategies. The
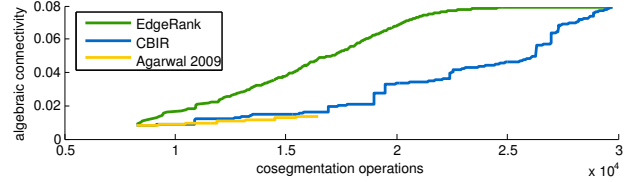
(a) Connectivity / Construction Time
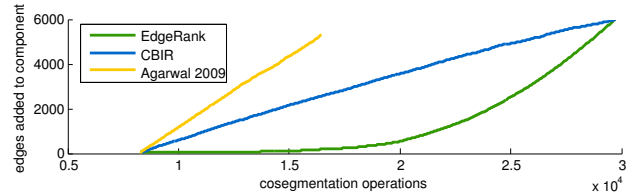
(b) Edges / Construction Time

(c) Connectivity / Edges

Figure 4. Comparison of Phase 2 strategies on the Art Museum collection with respect to construction time, number of edges added, and algebraic connectivity.

EdgeRank strategy improves the connectivity much more rapidly than CBIR or query expansion. The CBIR strategy can eventually achieve the same connectivity but requires many more cosegmentation operations. As shown in this figure, query expansion is not well suited for improving connectivity because it tends to add links in already well-connected areas of the graph. Figure 4(b) compares the strategies in terms of number of edges added to the component. Query expansion and CBIR add edges rapidly, successfully adding a new edge for roughly every 1.6 and 3.5 cosegmentation operations respectively. EdgeRank behaves very differently, adding very few edges initially while searching for the edges that are important for connectivity. Figure 4(c) shows that the EdgeRank strategy generates an image-graph with good connectivity after adding a relatively small number of edges compared with the other strategies. The connection between the algebraic connectivity and the graph structure can be seen in Figure 5. Edges in the graph constructed using EdgeRank are more uniformly distributed than in the graph constructed using query expansion.

3.4. Distributed implementation

Though constructing an Image Web is computationally expensive, the most expensive steps are independent and can be easily parallelized on a computer cluster. In our distributed implementation, a manager node issues feature ex-
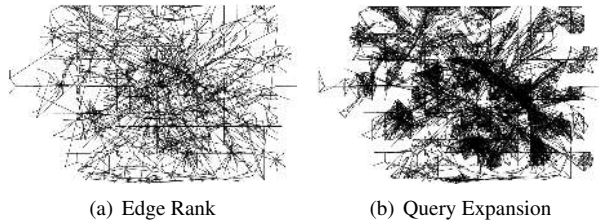
(a) Edge Rank   (b) Query Expansion

Figure 5. Image-graphs resulting from Phase 2 construction using EdgeRank (a) and Query Expansion (b) strategies respectively. The graph in (a) has a connectivity 5.5 times larger than the initial graph after adding 151 edges. The graph in (b) has a connectivity only 1.7 times larger after adding 5,212 edges.

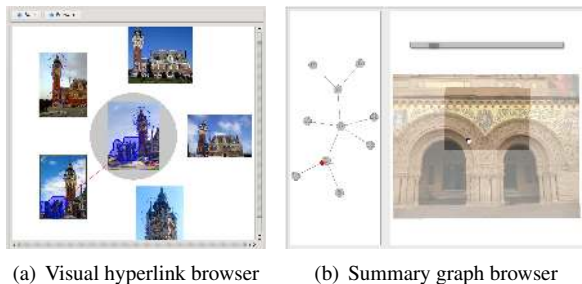(a) Visual hyperlink browser   (b) Summary graph browser

Figure 6. In (a), a visual hyperlink browser lets users navigate to related images by clicking on visual hyperlinks and provides a detailed view of a local neighborhood of the web. In (b), a summary graph browser provides a global view of the web allowing simple navigation around the entire web.

traction, CBIR, and cosegmentation jobs to worker nodes which compute and return the results. Table 1 shows some summary statistics. The reported construction times are for a cluster with 500 cores (2.33 GHz, 1GB RAM) and exclude the preprocessing steps for feature extraction and CBIR indexing.

4. Applications

In this section, we present applications for exploring image collections that exploit the connectivity captured in an Image Web at multiple resolutions. The first application provides a local view of the Image Web revealing details about the regions that link images. The second application provides a global view of the web that reveals the overall structure of the collection. The third application detects interesting relationships between images by identifying certain paths between densely connected regions of the web. Demonstrations of these applications are available at `http://geometry.stanford.edu/imagewebs`.

4.1. Exploration using visual hyperlinks

In this application, shared regions serve as *visual-hyperlinks* between images much like the text links between web documents. Figure 6(a) shows a graphical user interface built on this concept. The browser displays the cur-

rent image in the center, overlaid with the regions that link it to other images. In a ring around the center image, the browser shows other images that are related by a shared region. Hovering above a region highlights it and its corresponding region and draws a line connecting them. Clicking on a region follows the visual-hyperlink by selecting a new center image. The ring of thumbnail images serves as a summary of the neighborhood of related images while the visual hyperlinks indicate exactly which parts of the images are related. In a typical browsing session, a user may follow visual-hyperlinks associated with a particular object of interest, discover another interesting object, and then switch focus to explore the new object.

4.2. Exploration using a summary graph

Since Image Webs can be very large and complex, it is useful to produce a map or summary of the entire web to facilitate navigation. We have designed a novel algorithm which uses ideas from *persistent homology* [8] to compute a *summary graph* which captures global structure. In a summary graph, each image maps to either a vertex or to a position along an edge. Figure 6(b) shows an Image Web browser application based on summary graphs. The user can move forward or backwards along an edge of the summary graph to quickly flip through a set of connected images or choose which path to take at an intersection. The user's position is shown on the summary graph as she moves to help her see what part of the collection she is viewing and what parts she has not yet explored.

Computing a summary graph

Image webs can be viewed as noisy graphs where noise is introduced by the irregular sampling of views of the world and the limitations of image matching. Tools from the field of computational topology are appropriate in such a setting as they can detect persistent structures across multiple scales and are less sensitive to the exact notion of metric. We use persistent homology to remove insignificant features and to interpret our web as a stratified space consisting of one-dimensional strata joined at branching nodes. Such a model is appropriate for image collections captured by entities moving through a structured space like a city or building, and thus naturally containing long pieces of a linear character.

To compute a summary graph, we assign weights to the edges of an image graph by penalizing changes in viewpoint reflected in the relative scale and skew parameters of the affine transformations of matching regions between images. We then build a topological complex on which we will compute various topological invariants. The *Vietoris-Rips complex* $\mathcal{R}_\delta$ [12] is created from the image graph by picking a distance threshold δ and inserting, for all k, a k-

Collection Name (Source)	Images	Components (size > 1)	Largest Component	Construction Time (min)		
				Phase 1	Phase 2	Total
Stanford (Flickr)	193,277	12,505	11,240	173	96	269
Pittsburgh (StreetView)	50,224	23	49,907	7.9	70	78
London (Panoramio)	17,925	902	4,617	7.7	5.9	14
Art Museum (created)	1,257	5	1,217	0.06	0.74	0.8

Table 1. Summary statistics for Image Webs built from 4 different image collections. The construction times are for a computer cluster with 500 cores and a CBIR prefix of $M = 25$ in both Phase 1 and Phase 2.

simplex between nodes (images) if all pairwise distances are at most δ. As δ grows, we create a nested sequence of complexes, the *Rips filtration*, on which *persistent homology* can be applied [8]. The parameter δ represents the scale at which we choose to look at the graph and appropriate values can be chosen by examining the *persistence diagram* of the filtration, recording the lifetimes of topological features.

Next we assign a branching factor to each node N in a way that captures the local behavior of the graph around that node. Intuitively, this step works by continuously shrinking a ball centered at N, and tracking one-dimensional *relative homology groups* and their generators for the portion of the complex inside this ball [2]. We also relativize N with the boundary of this ball so that the number of one-dimensional homology classes matches our intuitive notion of the number of branches or degree at N. If more than two classes, with persistence at least ρ, die within a distance ε from N, we label N as a *branch node*; otherwise we label it as a *path node*. Note that in the absence of noise, the same effect can be achieved by simply considering the degree of each vertex. The parameters ρ and ε allow us to control the sensitivity of the method to noisy connections as well as short protrusions in the graph. Parameters ρ, ε are chosen by examining the persistence diagrams of local homology groups at few sample nodes. Persistent homology (and cohomology, for parametrizing the summary graph edges) is computed using the Dionysus C++ library for persistent homology [1]. Figure 7 shows the corresponding persistence diagram at a node N on a toy graph with unit edge weights.

Finally the branching factors computed for each node are used to partition the graph into connected components that correspond either to branching regions or path-like structures. An example Image Web and its summary graph are shown in Figure 8.
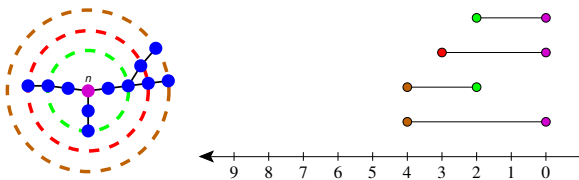
Figure 7. The persistence diagram for one-dimensional local homology groups at N. Birth/death points are colored-coded w.r.t balls of the corresponding radius.

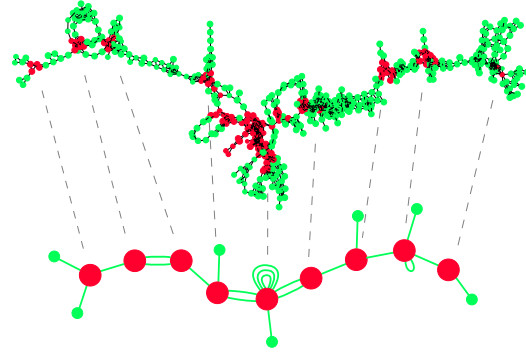1Dionysus library: http://www.mrzv.org/software/dionysus

Figure 8. An image graph (above) on 569 images taken on a tour of a college campus and its corresponding summary graph (below). Branches and paths are colored red and green respectively.

4.3. Discovering paths between landmarks

In this application we analyze the connectivity of the graph to identify densely connected subsets of the Image Web and the paths between them. In many data sets, densely sampled regions correspond to landmarks or objects people consider interesting. The paths between such landmarks are also interesting because they explain how they are related.

To analyze connectivity, we leverage the information captured by the Fiedler vector which was calculated during the construction of the Image Web as discussed in Section 3. The Fiedler vector is often used for partitioning vertices of a graph into tightly connected clusters. Here we use it to compute a connectivity score for edges in the graph via the same EdgeRank procedure that was used to compute the connectivity of potential edges in the graph. We identify the edges with high connectivity score (above a user adjusted threshold) and delete these edges to induce a partition on the vertices which correspond to landmarks. We choose the vertex with highest degree in each partition as the representative for the landmark. Finally, we compute the shortest paths between these vertices in the original Image Web. Since these paths must necessarily pass through the deleted "bottleneck" edges, they represent important connections between the landmarks.

Figure 9 shows the result of the connectivity analysis on an Image Web computed from a collection of 130 images taken around an office space. For this small collection, we can easily visualize the image-graph and highlight the high-connectivity edges as shown. We partitioned the graph into

Figure 9. An image graph overlaid with detected landmarks (orange) and shortest paths between them. Edges marked in red correspond to the bottleneck edges between landmarks. Two paths between landmarks correspond to physical paths in the office while one path is caused by a shared object (keyboards on both desks).

three distinct connected components which correspond to two different workspaces and the hallway between them. Figure 9 shows the representative landmark images and the images that lie on shortest paths between the landmarks.

5. Conclusion

In summary, this paper introduces the notion of capturing and exploiting global connectivity in large and dense image collections. We link images through image regions discovered by cosegmentation and use a measure of connectivity from spectral graph theory as a tool for improving the web construction process. We demonstrated a number of initial applications that exploit the global connectivity of such an Image Web. Just as the World Wide Web has established the value of interlinked text documents and the power of link analysis techniques, we believe there is value in having interlinked sets of images and in studying how to exploit their connectivity structure.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. *Proc. ICCV*, 2009.

[2] P. Bendich, D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and D. Morozov. Inferring local homology from sampled stratified spaces. In *Proc. FOCS*, 2007.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Proc. WWW*, 1998.

[4] O. Chum and J. Matas. Web scale image clustering. *Technical Report, Chech Technical University in Prague*, 2008.

[5] O. Chum, M. Perd'och, and J. Matas. Geometric minhashing: Finding a (thick) needle in a haystack. *Proc. CVPR*, 2009.

[6] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007.

[7] J. DiCarlo and D. Cox. Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11(8):333–341, 2007.

[8] H. Edelsbrunner and J. Harer. Persistent homology — a survey. *In Twenty Years After, eds. J. E. Goodman, J. Pach and R. Pollack, AMS.*, 453:257–282, 2008.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[10] P. Forssen. Maximally stable colour regions for recognition and matching. In *Proc. CVPR*, 2007.

[11] S. Gammeter, L. Bossard, T. Quack, and L. V. Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. *Proc. ICCV*, 2009.

[12] J.-C. Hausmann. On the Vietoris-Rips complexes and a cohomology theory for metric spaces. *Annals of Mathematics Studies*, 138:175 – 188, 1995.

[13] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, pages 304–317, 2008.

[14] Y. Jing and S. Baluja. Pagerank for product image search. In *Proc. WWW*, pages 307–316, 2008.

[15] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. *Proc. CVPR*, 2008.

[16] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. *Proc. ECCV*, 2008.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *Proc. ICCV*, pages pp. 91–110, 2004.

[18] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[19] D. Mosk-Aoyama. Maximum algebraic connectivity augmentation is NP-hard. *Oper. Res. Lett.*, 36(6), 2008.

[20] J. Philbin and A. Zisserman. Object mining using a matching graph on very large image collections. *Proc. ICVGIP*, 2008.

[21] I. Simon, N. Snavely, and S. Seitz. Scene summarization for online image collections. *Proc. ICCV*, 2007.

[22] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. In *Proc. SIGGRAPH*, 2008.

[23] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *Proc.SIGGRAPH*, 2006.

[24] H. Wang and P. Van Mieghem. Algebraic connectivity optimization via link addition. In *Proc. BIONETICS*, 2008.

[25] Y. Zheng, M. Zhao, Y. Song, and H. Adam. Tour the world: building a web-scale landmark recognition engine. *Proc. CVPR*, 2009.